An On-Board Processor for a Spaceborne Doppler Precipitation Radar: Requirements and Preliminary Design

S. L. Durden, M. A. Fischman, S. Tanelli, R. A. Johnson, A. Chu

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109

Abstract – **Use of Doppler velocity measurement in spaceborne precipitation radar is highly desirable, since it can allow more accurate retrieval of atmospheric latent heating, which depends on both cloud and rain microphysical processes and on dynamical processes, namely, vertical updrafts and downdrafts. However, if the rain within the antenna beam is very non-uniform, biases can result with conventional pulse-pair processing, and more sophisticated processing is required. The solution is a time-frequency approach, which uses the full Doppler spectrum at each point in time. Since precipitation radars normally operate continuously and collect large amounts of data, real-time, on-board processing of data is needed to reduce the data rate. The objective of this work is to develop an on-board data processor for spaceborne Doppler precipitation radar, using field-programmable gate array (FPGA) technology. This paper describes the simulations and analysis that have been completed during the project's first year. This includes both analytical calculations and bit-true simulations that allow the effects of finite word length to be explored. Tradeoffs have also been considered between speed and space within the chosen FPGA part. The results of these analyses have been used to develop a complete set of requirements for the processor. Initial design work has begun, based on these requirements. The design is being implemented in Verilog, and a description of this early design effort is also be given.**

I. INTRODUCTION

Doppler velocity measurements have played a key role in detecting and monitoring severe weather from ground-based and airborne radars [1]. Its use in spaceborne precipitation radar is highly desirable, since it can allow more accurate retrieval of atmospheric latent heating, which depends on both cloud and rain microphysical processes and on dynamical processes, namely, vertical updrafts and downdrafts [2]. Its capability in spaceborne radar has been incorporated into technology developed for a second-generation spaceborne precipitation radar (PR-2) [3]. An airborne PR-2 simulator has demonstrated real-time pulse compression and pulse-pair processing [4]. The spaceborne PR-2 design follows the successful Precipitation Radar (PR) on the Tropical Rainfall Measuring Mission (TRMM), which measures reflectivity but not Doppler velocity [5].

Because of the large platform motion, measurement of the precipitation Doppler velocity from space requires a large antenna and a high pulse repetition frequency (PRF) [6,7]. With these constraints, the Doppler mean can be estimated via the pulse-pair algorithm [1], if the rain is relatively uniform within the beam. However, if the rain is very non-uniform, biases can result [7], and more sophisticated processing is required. The solution is a time-frequency approach, which uses the full Doppler spectrum at each point in time, or position along track. The solution has been formulated in a way that is analogous to the Radon transform, the so-called combined frequency-time (CFT) technique. By calculating the Doppler spectrum as a function of time, the Doppler history of an individual target can be traced through time-frequency space. By processing the spectral signatures along these trajectories, rather than along lines of constant time, the technique deals with targets that have passed all the way through the beam, eliminating the bias of the instantaneous pulse-pair Doppler [8].

Since precipitation radars normally operate continuously and collect large amounts of data, real-time, on-board processing of data is needed. The objective of this work is to develop an on-board data processor for a spaceborne Doppler precipitation radar, using field-programmable gate array (FPGA) technology. The processor is to calculate the full Doppler spectrum, averaged over several range-bins. The resulting averaged spectra are then down linked for further processing. This paper describes the simulations and analysis that have been completed during the project's first year. This includes both analytical calculations and bit-true simulations that allow the effects of finite word length to be explored. Tradeoffs have also been considered between speed and space within the chosen FPGA part. The results of these analyses have been used to develop a complete set of requirements for the processor. Initial design work has begun, based on these requirements. The design is being implemented in the Verilog hardware description language. A description of this early design effort will also be given.

II. Processor Functional Requirements

Our prototype radar for development of the Doppler processor is the PR-2 noted above. However, the processor to be developed could also be applied to a simpler nadir-looking single frequency radar. The PR-2 radar design calls for a 5.3 m antenna and operation at Ku-band (13-14 GHz) and Ka-band (35-36 GHz) with PRF up to 6000 Hz. The system is a pulse-compression radar, transmitting a linear FM chirp with time-domain amplitude weighting. The bandwidth of the system is 4 MHz, providing roughly 60 m range resolution, with 30 m range bins, following the pulse compression operation. It is assumed that a data window extending from the surface to at least 14 km altitude would be recorded. Range ambiguity considerations allow a maximum pulse length of up to 33 microseconds. As currently envisioned, Doppler processing would only be performed on the Ku-band, co-pol channel; hence, the processor will be designed for operating on a single channel. Multiple processors could be used if Doppler processing were to be done on multiple channels. Although the PR-2 can scan, the CFT requires contiguous samples for processing. The Doppler data would likely be taken with either fixed antenna pointing or very limited scanning. A Doppler precision of 1 m/s is desired.

At each range bin the processor will multiply the complex radar data by a window that can be programmed. Next, an FFT will be calculated. This will be followed by the magnitude-squared operation and averaging over 8 range bins. The processor will handle 64 points along-track for each spectrum. The CFT estimation algorithm is not particularly sensitive to 64 vs. 128 point FFTs. N=64 gives fewer points in frequency and more points in time, while N=128 gives double the points in frequency and half the points in time. Either way, the frequency-time grid that is generated to estimate the Doppler centroid has the same number of points and comparable accuracy. The advantage of N=64 is that radix-4 or radix-8 schemes could be used if desired, in addition to the more common radix-2. The 64 pulses are acquired in 11 ms at a PRF of 6000. Thus, real-time operation requires that all operations (including input and output) on 64 pulses be completed within 64/(6 kHz) ≈ 11 ms.

Assuming sequential application of the processor to each range bin, the time available for the FFT is just under 11 ms/480 bins, or about 22 μs; 480 30-m range bins are assumed, giving a window of 14.4 km. This also assumes that nearly the full 11 ms is available for FFT processing, meaning that the input and output are pipelined and that the time for the other operations is very small compared with the FFT.

One of the key requirements for the processor is dynamic range. Within a single range bin the output Doppler spectrum must accommodate a dynamic range of 30 dB. This is to allow an accurate estimate of the centroid from the spectrum. The processor shall accommodate 30 dB dynamic range over the range window, based on 25 to 55 dBZ rainfall radar reflectivity. It is assumed that CFT processing is not needed for the surface return; pulse-pair processing can be used for the surface. For reflectivities lower than 25 dBZ, thermal noise will limit the Doppler spectrum to less than 30 dB of dynamic range. The overall dynamic range requirement for the processor is thus 60 dB.

As derived in Oppenheim and Schafer [9] the signal-to-noise ratio after an FFT operation (assuming noiseless input) is $2^{2b}/4N$, where b is the number of unsigned bits and N is the length of the FFT. In dB, this is 6.02b-10log(4N). This analysis assumes a scaling of 0.5 (one bit shift right) at each stage. For 16-bit signed arithmetic, b=15. Using N=64, the predicted S/N is 66 dB which just meets the required dynamic range. Using 23 bits, the predicted S/N is 114 dB, which is well beyond requirements. Moving in the other direction, b=7 or 11 does not meet dynamic range requirements.

Note that weather echoes have in-phase and quadrature (I and Q) voltages that are normally distributed; hence, the root mean square (rms) voltage needs to be set at least 6 dB below the maximum voltage, reducing the available dynamic range by 6 dB. If the rms voltage is too large, relative to the full-scale value, several samples in the input to each FFT will likely be saturated, degrading the performance. Use of b=15 bits thus gives a dynamic range of 60 dB, just meeting the requirement.

During the complex phasor rotation operation in the FFT butterflies, it is possible for either the real or imaginary component of the butterfly output to overflow (to fall outside of the $\pm 2^{15}$ boundaries), even with the application of the 0.5-scaling factor. To prevent internal overflows from occurring, the registers within each butterfly will be designed with an extra MSB cushion bit. In terms of the FFT's interface to other processing blocks in the algorithm, the data I/O will still use a 16-bit signed integer standard for real and imaginary components. However, at the level of the butterfly module, the data I/O widths will be designed to accommodate 17-bit signed integer values (b=16).

To compute the magnitude squared of the complex spectrum at each range bin. 17-bit I / 17-bit Q input data will be expanded to 33-bit unsigned integer format for the I^2+Q^2 output values. These are truncated to 29 bits. The selection of the 29 bits out of the 33 bits is to be programmable by loading a 3-bit register. The 29-bit

spectra are then summed over 8 range bins; this expands the bit width to 32-bit unsigned integer format. The output of the processor is the averaged Doppler spectra at each of 60 range bins (480 input bins, reduced by 8-fold averaging).

III. Doppler Processor Design

A. Overview

The objective in the first year of the AIST-2 project is to develop the algorithm for the Doppler Precipitation Radar processor and begin mapping the functions of the algorithm into a single field-programmable gate array device. The device selected is the 1 million gate Xilinx Virtex-1000 FPGA (XCV1000-4BG560), which provides a high logic density for implementing the functions of the processor in real-time and has an equivalent radiation-tolerant part for the space environment. The V-1000 is also a convenient choice for leveraging off of the hardware platform previously developed in AIST-1, since it was a common FPGA part for the spaceborne PR-2 pulse-compression processor/controller [10]; by reusing this hardware platform with a newly designed V-1000 configuration file, the Doppler rain processor can be demonstrated in a laboratory environment during the third year of this project.

A functional block diagram of the FPGA-based Doppler processor design is shown in Fig. 1. All of the logic modules in the FPGA will be designed to run synchronously off of a common 20 MHz system clock. The input signal consists of 16-bit I / 16-bit Q complex baseband range bin samples coming from the pulse-compression part of the PR-2 processor at 5 Msample/s (160 Mbit/s peak rate) via a bidirectional local address and data (LAD) bus. Over the 11 ms processing interval, the received I/Q data from 64 radar pulses are stored in external static RAM in rows (representing the 480 range altitude bins) and columns (representing the 64 successive radar pulses along-track). After an entire processing interval is captured in memory, an SRAM driver circuit reads the data back into the processor row-by-row in a "corner turn" fashion so that Fast Fourier Transform processing can be performed on range bin samples at each altitude. 64-point complex data is then read into the main processing core, which consists of an amplitude windowing operation, the 64-point FFT module, power detection, and a factor-of-8 averaging in range. As mentioned in the Functional Requirements section, the power spectrum output result for each range bin must be available from the processing core within 22 μs after processing has started in order to guarantee real-time operation. When the averaged Doppler spectral data is ready, it is written to a second bank of external SRAM with the lower 6 offset address bits reversed (so that the FFT output data is rearranged in natural order, from 0 to 6 kHz), and is subsequently read out to a computer via the LAD bus. Given an output data resolution of 32 bits per frequency bin, 64 bins per spectrum, and 60 total averaged range bins, the science data volume output per processing interval is reduced to 11.5 Mbit/s.

B. Bit-true analysis of algorithm

To analyze how the Doppler processing algorithm could be implemented in FPGA hardware, a bit-true model of the processor was developed in Simulink using the "Fixed-Point Blockset" library. This library includes fixed-point adders and multipliers which emulate the actual effects of digital signal processing hardware such as numeric overflows or quantization (roundoff) errors. The fixed-point Simulink design could then be interfaced as the device-under-test to a Doppler rain spectra simulator, developed in MATLAB, to verify whether the processor requirements are met in terms of dynamic range and vertical velocity measurement precision.

The key component in the fixed-point Doppler processor design is the N=64 point, radix-2 FFT core that was discussed previously. A decimation-in-frequency approach was chosen for the FFT which consists of $\log_2 N = 6$ stages, each stage containing $N/2 = 32$ adder/multiplier butterfly operations. In all, 192 butterfly operations are thus required to complete one FFT operation. The butterfly architecture uses 16-bit I/Q resolution on the twiddle factor and 17-bit resolution on the complex input and output data pair to prevent overflows during the complex phasor rotation (multiplication) by the twiddle factor. Fig. 2 shows the hierarchical design entered for the bit-true FFT module in Simulink, starting with the top-level design (six radix-2 stages plus a reordering stage at the end), going down in level to one of the stages (containing 32 butterflies), and finally going to a single 17-bit butterfly with a twiddle factor constant.

Zrnic's method [11] was used to generate realistic rain echo signals as an input to the Doppler processor design for testing. If $x_0, x_1, \ldots, x_{N-1}$ represent the 64 return samples measured at a given range altitude for the rain scatterers as the satellite moves along track, then these samples will typically have a Rayleigh distributed envelope and uniformly distributed random phase. Also, the Doppler spectrum $X_0, X_1, \ldots, X_{N-1}$ of the 64 samples will have a power spectrum which is approximately Gaussian-shaped (determined by the antenna pattern) and which has a center frequency f_0 proportional to the rain's vertical velocity:

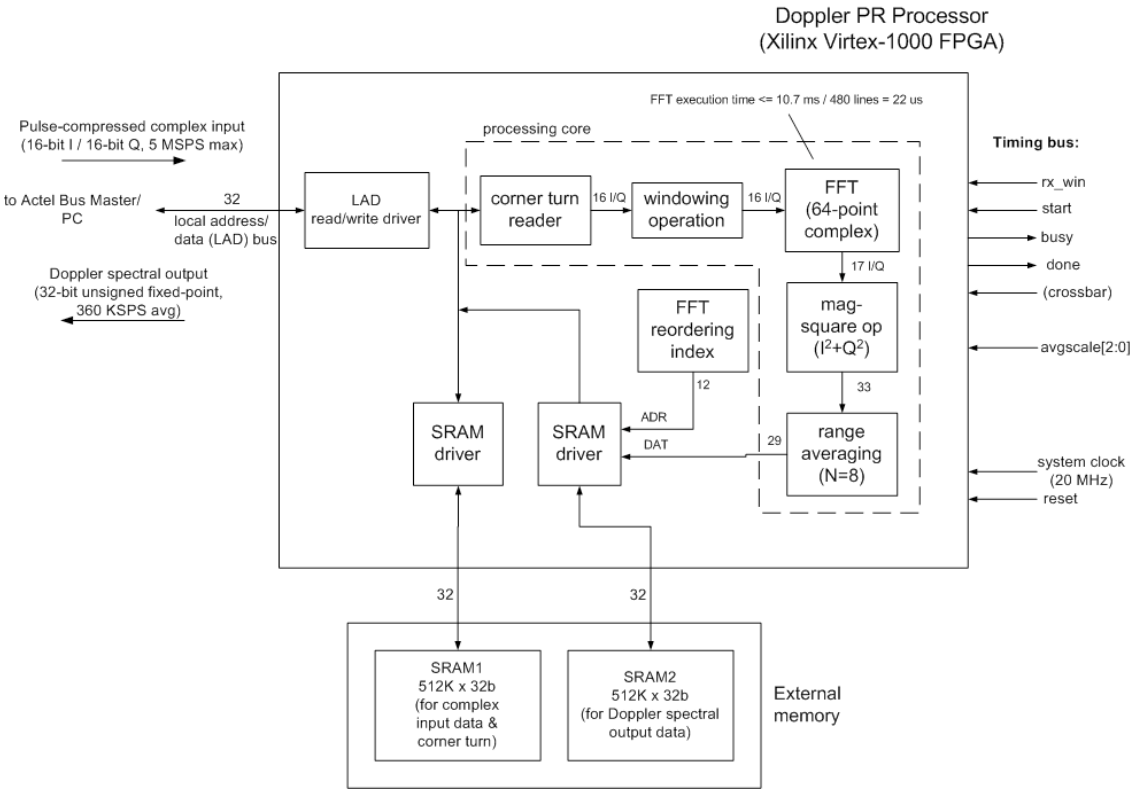

Fig. 1. Functional top-level block diagram of the Doppler precipitation radar processor design, implemented onto the Xilinx Virtex-1000 FPGA.

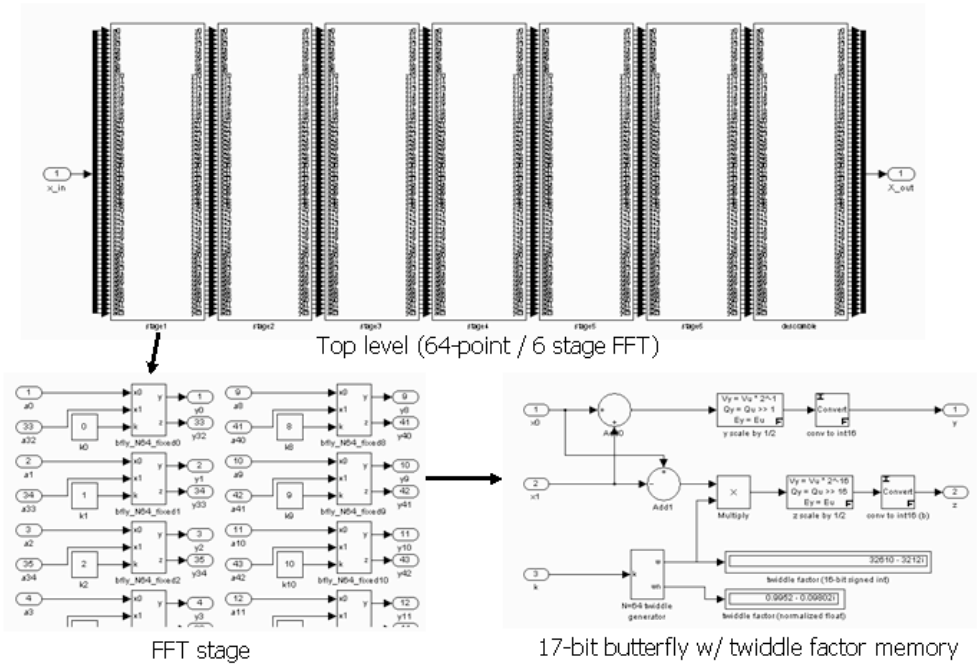

Fig. 2. Hierarchical FFT fixed-point algorithm designed in Simulink.

$$\left\langle |X[k]|^2 \right\rangle = A\left[e^{-\frac{1}{2}\left(\frac{kf_s/N - f_0}{\sigma_f}\right)^2} + e^{-\frac{1}{2}\left(\frac{f_s - kf_s/N - f_0}{\sigma_f}\right)^2} \right] \qquad (1)$$

$$k = 0,1,\ldots,N-1,$$

where f_s = 6 kHz for the pulse repetition frequency, σ_f is the standard deviation (bandwidth) of the Doppler spectrum, and A is proportional to the return signal strength. The two exponential terms in (1) represent the positive and negative Doppler frequency components measured by the nadir-looking antenna as the satellite moves forward. Both the positive and negative Doppler frequency components are aliased in each of the 64 frequency bins due to the finite sampling rate f_s.

Equation (1) represents the expected Doppler power spectrum for rain. Once this is known, multiple radar looks of the rain target can be simulated by multiplying the X values by an exponentially distributed random variable, taking the square root to get a proportional voltage value, and then multiplying by a complex phasor $e^{j\theta}$ with uniform random phase θ. This result is then inverse Fourier transformed to generate statistically independent, time-domain samples for x_n, n = 0,...,N-1.

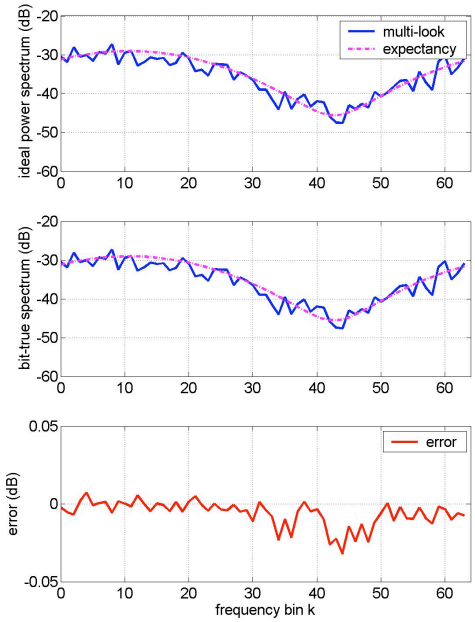

Fig. 3. Processor simulation for a Doppler rain target with a vertical velocity of ~11 m/s (f_0 = 1 kHz). Graphs from top to bottom show the output responses for: the ideal (floating-point) model of the processor core; the bit-true (fixed-point) model of the processor core; and the error signal (difference between bit-true and ideal responses). The plotted power spectra are averaged over 8 independent radar looks of the rain target to simulate averaging in range.

Time-domain Doppler rain echo data were generated and input to the fixed-point Simulink design for a typical case with a Doppler centroid frequency of f_0 = 1 kHz (a vertical velocity of $f_0\lambda_0/2$ = 11 m/s), a spectral standard deviation of σ_f = 1 kHz, and N_l = 8 independent radar looks. Fig. 3 compares the output results between an idealized floating-point model of the processor and the bit-true model developed in Simulink. (The input signal strength was set to 10 dB below full scale, where full scale is defined as a signal power of $\langle|x|^2\rangle$ = 1/3 and the maximum digital value is normalized to unity [9].) Qualitatively, the floating-point and fixed-point spectral outputs agree closely with each other and match the expected Gaussian power spectrum to within 2-3 dB. In terms of quantization error effects, the power density measurement error is at most 0.02-0.03 dB across the spectrum, which is negligible compared to the radar fading noise error of $1+1/\sqrt{N_l}$ 1.3 dB.

A large number of output test vectors from the Simulink processor model were also generated and input to the CFT algorithm's Doppler centroid estimator to see if there were any noticeable fixed-point biasing effects. One-thousand 64-point, full-scale Doppler outputs were generated at each center frequency f_0 = 0, 500, 1000, and 2000 Hz, again with σ_f = 1 kHz. It was found that for a range averaging factor of N_l=8, the bias in the mean frequency of the spectrum (compared to the ideal spectrum generated by floating-point software) was no more than 3 Hz error for all four center frequency cases. This means that the fixed-point processor hardware introduces no more than ~3 cm/s error in the vertical rain velocity measurements and thus satisfies the 1 m/s measurement precision requirement.

Dynamic range was another figure-of-merit tested in the bit-true processor model. This was tested by varying the Doppler rain spectra over a range of signal strengths and measuring the output signal-to-noise plus distortion (SINAD) ratio, defined as the ratio of the floating-point signal power out,

$$P_S = \sum_{k=1}^{N} |X_{ideal}(k)|^2 \qquad (2)$$

to the noise power out due to fixed-point errors,

$$P_N = \sum_{k=1}^{N} |X_{fixed}(k) - X_{ideal}(k)|^2 . \qquad (3)$$

Simulations show that a linear region occurs in the response which extends over approximately a 60 dB range, bounded by the quantization noise floor at very low signal levels (65 dB down from the full scale 16-bit input) and by numeric saturation at high signal levels (5 dB below full scale). This 60 dB simulated range agrees closely with the theoretical dynamic range formulation of $6.02b\text{-}10\log_{10}(4N) = 66$ dB presented earlier for the FFT operation. With this performance, the processor design will have more than sufficient dynamic range for rain reflectivities of interest in the 25 to 55 dBZ range.

C. FFT Verilog core design

As part of the first year effort, we have started designing the FFT component of the Doppler processor algorithm in the Verilog hardware description language (HDL). A preliminary version of the 64-point complex FFT core has already been coded, tested in behavioral simulations, and run through logic synthesis for the V-1000 FPGA. Based on timing constraint estimates, the 22 μs execution time limit for one FFT operation could be satisfied by instantiating just one butterfly module and running all 192 data passes through this single butterfly. This method was pursued in order to save logic resources aboard the FPGA chip.

Fig. 4 illustrates the hardware architecture and Verilog design for the 64-point FFT. This design is based on the Singleton technique [9], which uses multiplexer switches to permute the data order into the butterfly and uses two banks of first-in/first-out (FIFO) memory to store intermediate results. The twiddle factor coefficients, $\Omega_N^k = e^{-j2\pi k/N}$, are stored in the FPGA's lookup table (LUT) hardware and are addressed in sequence for the 192 butterfly passes. As the final 32 passes are completed, a register is enabled to send the Fourier transformed result out in a data burst to the next processing block. (Note as a consequence of the data permutation, the frequency data is output in a bit-reversed address sequence.)

The single-butterfly FFT core was designed at the top-level in Verilog and used pre-existing IP cores designed by Xilinx at a lower level for generic components such as multipliers. After logic synthesis, the design summary showed that the FFT occupied ~9% of the available configurable logic blocks (CLB) on-chip, which leaves a comfortable area margin for other functions of the algorithm. Behavioral simulations with the ModelSim design tool showed that the 64-point transformed result becomes available 390 system clock cycles (19.5 μs) after the input data is received. While this processing time satisfied the 22 μs real-time requirement, it left little margin for the other operations within the Doppler processor. To relax the timing margin, the design is currently being modified slightly from the depiction in Fig. 4 by adding a second butterfly instance, operating in parallel with the first, to gain approximately a factor-of-2 improvement in speed. (This change will increase the occupied CLB real-estate on the V-1000 from 9% to approximately 20%.)

In terms of functionality, the FFT Verilog hardware was simulated for several basic test vector types (zero-lag impulse response, dc signal response, and time-delayed impulse response), and the outputs were compared to those of the fixed-point Simulink model. For all three cases there was a bit-for-bit agreement between the fixed-point FFT model and the Verilog simulation. The next step in the HDL verification will be to insert test vectors generated from the Doppler rain spectra software into the Verilog simulation of the FFT to confirm the dynamic range and measurement precision.

IV. Conclusions and Future Plans

We have described requirements and preliminary design of a Doppler processor for a spaceborne precipitation radar. The requirements have been verified via simulation. The processor block-level design has also been presented and Implementation of some of the blocks has begun. We plan to continue Verilog coding, with the goal of completing the Verilog description in the fall of 2004, although testing via simulation will extend into 2005. Late in 2004 we also plan to begin preparations for testing the processor in a custom board with a Xilinx Virtex FPGA.

Acknowledgment

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

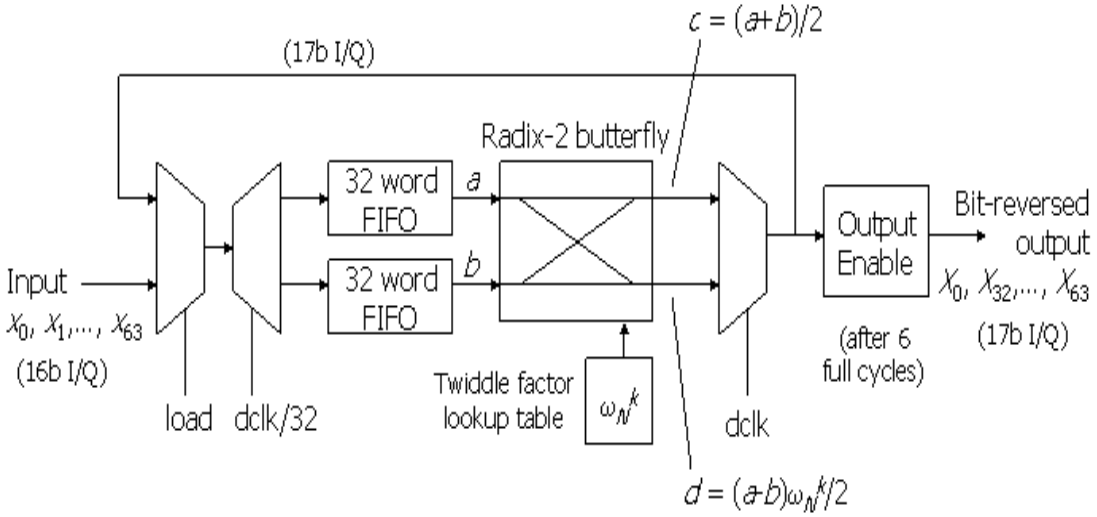

Fig. 4. FPGA hardware architecture implementation of the 64-point FFT core.

REFERENCES

[1] R. J. Doviak and D.S. Zrnic, *Doppler Radar and Weather Observations, 2nd Edition*. New York: Academic Press, 1993.

[2] S. A. Braun and R. A. Houze, "The heat budget of a midlatitude squall line and implications for potential vorticity production," *J. Atmos. Sci*., vol. 53, pp. 1217-1240, 1996.

[3] E. Im and S. L. Durden, "Dual-frequency, wide-swath radar for spaceborne rainfall measurements," *Specialist Meeting on Microwave Remote Sensing*, Boulder, CO., 2001.

[4] G. A. Sadowy, A. C. Berkun, W. Chun, E. Im, and S. L. Durden, "Development of an advanced airborne precipitation radar," *Microwave J*., vol. 46, pp. 84-98, January 2003.

[5] T. Kozu et al., "Development of Precipitation Radar onboard the Tropical Rainfall Measuring Mission (TRMM) satellite," *IEEE Trans. Geosci. Remote Sensing*, vol. 39, pp. 102-116, 2001.

[6] P. Amayenc, "Proposal for a spaceborne dual-beam rain radar with Doppler capability," *J. Atmos. Oceanic Technol*., vol. 10, pp. 262-276, 1993.

[7] S. Tanelli, E. Im, S. L. Durden, L. Facheris, and D. Guili, "The effects of nonuniform beam filling on vertical rainfall velocity measurements with spaceborne Doppler radar," *J. Atmos. Oceanic Technol.*, vol. 19, pp. 1019-1034, July 2002.

[8] S. Tanelli, E. Im, S. L. Durden, L. Facheris, D. Guili, and E. A. Smith, "Rainfall Doppler velocity measurements from spaceborne radar: Overcoming NUBF effects," *J. Atmos. Oceanic Technol*., vol. 21, pp. 27-44, January 2004.

[9] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1975.

[10] M.A. Fischman, A.C. Berkun, W. Chun, and E. Im, "Design and verification of the Second-Generation Precipitation Radar processor/controller," in *Proc. Earth Sci. Tech. Conf.*, NASA Earth Science Technology Office, 2003.

[11] D.S. Zrnic, "Simulation of weatherlike Doppler spectra and signals," *J. Applied Meteor.*, vol. 14, pp. 619-620,1975.